

МІНІСТЕРСТВО ОСВІТИ І НАУКИ,  
МОЛОДІ ТА СПОРТУ УКРАЇНИ

Національний технічний університет  
«Харківський політехнічний інститут»

**МЕТОДИЧНІ ВКАЗІВКИ**  
**до лабораторної роботи**  
**«Комбінований тип даних у Delphi»**  
з курсу «Програмування»  
для студентів напрямку 6.040302 – Інформатика  
(спеціалізація «Соціальна інформатика»)

Затверджено редакційно-видавничою  
радою університету,  
протокол № 2 від 01.12.10.

Харків НТУ «ХПІ» 2011

Методичні вказівки до лабораторної роботи «Комбінований тип даних у Delphi» з курсу «Програмування» для студентів напрямку 6.040302 – Інформатика (спеціалізація «Соціальна інформатика») / Уклад. М. І. Безменов. – Х. : НТУ «ХП», 2011. – 13 с.

Укладач М. І. Безменов

Рецензент Л. М. Любчик

Кафедра системного аналізу і управління

## ВСТУП

Записи є типом даних який дозволяє поєднувати о удному об'єкті-змінній дані різних типів. Саме з записи є тим типом, який о подальшому породив об'єктний тип і класи.

**Метою** даної лабораторної роботи є освоєння методики комбінованих даних (записів) у програмах, написаних мовою Delphi.

## 1. ТЕОРЕТИЧНІ ОСНОВИ

### 1.1. Звичайні записи

У Delphi існує спеціальний комбінований тип даних, що дозволяє поєднувати відразу кілька типів. Такий тип даних називають записом. Формат опису змінної типу «запис» наступний:

```
var
    ім'я_запису: record
        ім'я1: тип1;
        ім'я2: тип2;
        ...
        ім'яN: типN;
    end;
```

Частини запису, призначені для безпосереднього зберігання інформації (ім'я1, ..., ім'яN), називаються полями запису. Для кожного з полів повинні бути зазначені ім'я та тип. Якщо кілька полів мають один тип, то їх можна оголосити спільно, перелічивши імена через кому.

У програмі можна, наприклад, оголосити таку змінну типу «запис»:

```
var
    Stud: record
        Surname: string[20];           // Прізвище
        AverageEval: Real;             // Середня оцінка
    end;
```

Опис запису завершується службовим словом **end**, якому відповідає службове слово **record** (але не **begin**).

Рекомендується спочатку оголосити ім'я типу запису, яке надалі буде використовуватися при описі змінних:

```

type
  TStud = record
    Surname: string;
    AverageEval: Real;
  end;
var
  Stud: TStud;

```

Щоб звернутися до поля запису, необхідно не просто вказати ім'я поля, але перед ним зазначити ім'я запису, з'єднавши їх крапкою:

```

Stud.Surname := 'Супченко';
Stud.AverageEval := 4.8;

```

Альтернативою такому зверненню є звертання з використанням оператора приєднання **with**:

```

with ім'я_запису do оператор;

```

Оператор **with** автоматично приєднує зазначене в ньому ім'я\_запису до всіх імен, які входять в оператор, що стоїть після **with**, і збігаються з іменами полів запису. До імен, що не збігаються з іменами полів запису, приєднання не проводиться. Якщо приєднання потрібно виконати в декількох операторах, то їх за допомогою **begin–end** поєднують у складений оператор. Наприклад, наведені вище оператори можна записати так:

```

with Stud do begin
  Surname := AnsiUpperCase(Surname);
  AverageEval := 0;
end;

```

Записи можуть бути елементами масиву. Наприклад, для наведений тип TStud можна використати для опису масиву:

```

var
  Group = array [1..35] of TStud;

```

У цьому разі для звертання для полів записів крапку зазначають після квадратних дужок, у які укладено індекси; наприклад: group[i].Surname.

Поля запису можуть мати будь-який тип. Наприклад, як поле запису можливий масив

```

type
  SessionRec = record
    AverageEval: Real;

```

```

    Eval: array[1..5] of 2..5;
end;

```

або запис

```

type
  StudRec1 = record
    Surname: string;
    Session: SessionRec;
  end;
var
  Report: array[1..100] of StudRec1;

```

В останньому випадку для звертання до полів запису доводиться будувати цілий ланцюжок приєднань.

Можна також указати через кому список імен, що приєднуються, в операторі **with**. При цьому приєднання починається з імені, що стоїть правіше в списку. Так, наступні чотири оператори еквівалентні:

- 1) **report**[17].Session.Eval[3] := 4;
- 2) **with** Report[17], Session **do** Eval[3] := 4;
- 3) **with** Report[17] **do**  
     **with** Session **do** Eval[3] := 4;
- 4) **with** Report[17].Session **do** Eval[3] := 4;

Можливість використання записів як полів інших записів дозволяє організовувати оперування складними структурами різнорідних даних.

Щоб оголосити типізовану константу записного типу, необхідно після імені константи слідом за знаком рівності в круглих дужках перелічити імена полів, зв'язавши їх зі значеннями двокрапкою. При цьому як роздільник вживається крапка з комою. Якщо полем запису є масив або запис, то задавання значення для такого поля виконується у своїх круглих дужках. Це ж має місце й у випадку типізованих констант – масивів з елементами записного типу.

Наприклад:

```

type
  SessionRec = record
    AverageEval: Real;
    Eval: array[1..5] of 2..5;
  end;
  TVariants = array [1..3] of SessionRec;
const
  Etalon = (AverageEval: 5.0; Eval = (5, 5, 5, 5, 5));
  Variants: TVariants =

```

```
((AverageEval: 2.0; Eval = (2, 2, 2, 2, 2)),
(AverageEval: 3.0; Eval = (3, 3, 3, 3, 3)),
(AverageEval: 4.0; Eval = (4, 4, 4, 4, 4)));
```

## 1.2. Записи з варіантами

Крім звичайних записів, у Delphi є можливість використання записів з варіантами, ідея яких полягає у визначенні в рамках однієї структури декількох різних (альтернативних) записів. Нехай, наприклад, необхідно організувати обробку інформації про книги (автори, назва книги, місце видання, назва видавництва, рік видання) і журнальні статті (автори, назва статті, журнал, номер, рік видання). Якщо в одному масиві необхідно зосередити інформацію про такі різномірні записи, потрібно в одному записі описати всі можливі поля і якимось чином обмежити доступ до тих з полів, які не вживаються в конкретному запису.

Запис з варіантами складається з двох частин – фіксованої, у якій перелічуються спільні для всіх альтернатив поля, і змінної (варіантної), структура якої змінюється залежно від значення особливого поля, так званого поля тегу (ознаки). Для поставленої вище задачі можна запропонувати таку структуру запису:

```
type
  TCatalog = record
    Author: string[20];
    Title: string[100];
    Year: Integer;
    case Key: Byte of
      0: (Town: string[20];
          Publishing: string[20]);
      1: (Journal: string[20]; Num: Byte);
    end;
var
  Catalog: array[1..100] of TCatalog;
```

Особливості записів з варіантами:

- варіантна частина може бути тільки одна, і вона повинна бути останньою;
- варіантна частина подібна з оператором **case**, у якого відсутнє службове слово **end**; замість **end** для **case** використовується **end** для **record**;

- полем тегу є ім'я (власне тег) і тип, що стоять за **case**;
- поле тегу може складатися тільки з типу без вказівки імені;
- тег може мати тільки дискретний тип;
- тег входить у фіксовану частину запису і завершує її;
- опис полів, що стосуються окремих значень тегу, беруть у круглі дужки;
- якщо варіант для якого-небудь значення тегу не потрібен, то він задається порожнім полем, укладеним у круглі дужки;
- якщо яке-небудь з полів у варіантній частині – це теж варіантний запис, то воно повинне бути останньою частиною варіанта.

При оголошенні типізованих констант як записів з варіантами подають значення тільки одного з варіантів.

Варіантні записи, полегшуючи оперування даними, у той же час збільшують обсяг використаної пам'яті, оскільки під такого роду змінні завжди виділяється пам'ять, необхідна для зберігання найдовшого запису. Контроль за тим, який з варіантів реалізований у даній змінній, повністю покладається на програміста (можна, наприклад, записати в деяку змінну `Catalog[i]` інформацію про книгу, а обробляти її як інформацію про журнальну статтю). Тому при роботі з варіантними записами треба завжди враховувати значення тегу.

Оскільки записи звичайно містять досить великі обсяги різномірної інформації, вводити їх із клавіатури частіш за все проблематично (річ йде, природно, не про окремі записи, а про їх послідовності). У цьому випадку на допомогу приходять типізовані файли, базовим типом яких є відповідний записний тип.

## 2. ПРИКЛАД ПРОГРАМИ

Дано натуральне число  $n$  та інформація про прізвища та середній бал  $n$  студентів. Вивести прізвища тих студентів, чий середній бал вищий від загального середнього бала.

**Розв'язання.** Для розв'язання задачі створимо форму, розмістивши на ній два однорядкових редактори з мітками над ними, багаторядковий редактор та звичайну кнопку і здійснивши такі зміни значень деяких властивостей цих компонентів:

Перша мітка:

Name — `lbOutput1`

Друга мітка:

Name — `lbOutput12`

Visible — `False`

Перший однорядковий редактор:

Name — edInput1

Text — 1

Другий однорядковий редактор:

Name — edInput2

Text — очистимо

Visible — False

Багаторядковий редактор:

Name — mmOutput1

Lines — очистимо

Кнопка:

Caption — Увести

У розділі опису типів інтерфейсної частини модуля перед описом класу TForm1 опишемо такий записний тип:

```
TStudRec = record
  Surname: string;
  AverageEval: Real;
end;
```

У секції **private** (або **public**) опису класу TForm1 опишемо такі поля:

```
Count: Integer;
StudArray: array of TStudRec;
```

а на початку розділу **implementation** слідом за описом форми розмістимо опис змінної *c* для підрахунку кількості студентів:

```
var
  c: Integer = 0;
```

Для розв'язання задачі можливе застосування таких двох опрацьовувачів подій:

```
procedure TForm1.Button1Click(Sender: TObject);
var
  Sum, ArithmeticMean: Real;
  i: Integer;
begin
  if Count = 0 then begin
    Count := StrToInt(edInput1.Text);
    SetLength(StudArray, Count);
```



```

mmOutput1.Lines.Add('Усього студентів '
                    + IntToStr(Count));
lbOutput1.Caption := 'Уведіть прізвище';
lbOutput2.Caption := 'Уведіть середній бал';
edInput2.Text := '5';
lbOutput2.Visible := True;
edInput2.Visible := True;
edInput1.SetFocus;
end
else begin
    //Нижче використовується оператор приєднання
    with StudArray[c] do begin
        Surname := edInput1.Text;
        AverageEval := StrToFloat(edInput2.Text);
        mmOutput1.Lines.Add(surname + ' '
                            + FloatToStr(AverageEval));

        Inc(c);
    end;
    if c < Count then
        edInput1.SetFocus
    else begin
        Sum := 0;
        for i := 0 to Count - 1 do
            Sum := Sum + StudArray[i].AverageEval;
        ArithmeticMean := Sum / Count;
        mmOutput1.Lines.Add('Результат:');
        for i := 0 to Count - 1 do
            if StudArray[i].AverageEval > ArithmeticMean then
                mmOutput1.Lines.Add(StudArray[i].Surname);
        Button1.Visible := False;
        edInput1.Visible := False;
        lbOutput1.Visible := False;
        edInput2.Visible := False;
        lbOutput2.Visible := False;
    end;
end;
end;

procedure TForm1.FormCreate(Sender: TObject);
begin

```

```
lbOutput1.Caption := 'Уведіть кількість студентів';  
Count := 0;  
end;
```

### 3. ЗАВДАННЯ НА ЛАБОРАТОРНУ РОБОТУ

За час, відведений для виконання лабораторної роботи (2 академічні години), студент повинен:

1. Розробити алгоритм розв'язання задачі, запропонованої для програмування.
2. Здійснити проектування форми для функціонування розроблюваної програми.
3. Здійснити програмну реалізацію розробленого алгоритму.
4. Здійснити відлагодження програми, виправивши синтаксичні та логічні помилки.
5. Підібрати тестові дані для перевірки програми, включаючи виняткові випадки.
6. Оформити звіт до лабораторної роботи.
7. Відповісти на контрольні запитання.
8. Здати викладачу працездатну програму з демонстрацією її роботи на декількох варіантах вихідних даних.

### 4. ВАРІАНТИ ЗАДАЧ

1. Дано натуральне число  $n$  і послідовності дійсних чисел  $a_1, a_2, \dots, a_n; x_1, x_2, \dots, x_n; y_1, y_2, \dots, y_n$ . Ці дані визначають на координатній площині  $n$  квадратів зі сторонами, паралельними координатним осям:  $a_i$  — сторона квадрата,  $(x_i, y_i)$  — координати центра  $i$ -го квадрата,  $i = 1, 2, \dots, n$ . Чи існують на координатній площині точки, що належать усім квадратам? Якщо так, то зазначити координати однієї з них.
2. Дано натуральне число  $n$  і множину з  $n$  точок на площині, що задаються парами дійсних чисел  $(x_i, y_i)$ ,  $i = 1, 2, \dots, n$ , а також пару дійсних чисел  $(u, v)$ , які визначають координати додаткової точки. Знайти номер точки множини, що є:
  - а) найближчою до точки  $(u, v)$ ;
  - б) найвіддаленішою від точки  $(u, v)$ .

3. Множина точок задається парами дійсних чисел. Знайти точку цієї множини, для якої:
- а) сума відстаней до інших точок множини мінімальна;
  - б) сума відстаней до інших точок множини максимальна;
  - в) сума відстаней до центра множини мінімальна;
  - г) сума відстаней до центра множини максимальна.
4. Множина точок задається парами дійсних чисел. Знайти дві різні точки цієї множини, відстань між якими:
- а) мінімальна;
  - б) максимальна.
5. Дано дві множини точок  $A$  і  $B$ , задані парами дійсних чисел. Множина  $A$  містить  $m$  точок, а множина  $B$  –  $n$  точок, де  $m$  і  $n$  – задані натуральні числа. Знайти відстань між множинами  $A$  і  $B$ . Відстань між двома множинами точок визначати як:
- а) відстань між найближчими одна до одної точками цих множин;
  - б) відстань між найвіддаленішими точками цих множин;
  - в) відстань між центрами цих множин.
- У двох перших випадках знайти також номери і координати точок, за якими визначалася відстань між множинами.
6. Дано масив з 10 дат (число, місяць, рік). Зазначити всі літні дати, а також найбільш ранню дату.
7. Маємо каталог на  $n$  книг і журнальних статей ( $n \leq 100$ ). Вивести інформацію про публікації, видані після 1990 р. (окремо книги і статті).
8. Дано інформацію про оцінки, отримані студентами академічної групи під час сесії. Інформація подається у такому вигляді: прізвище студента (рядок не більше 30 символів), форма навчання (бюджетна або контрактна – одна з латинських літер В або С), 5 оцінок (цілі числа 2, 3, 4, 5 або 0, причому 0 відповідає випадку, коли студент не складав відповідний іспит). Має бути також передбачена можливість наявності інформації про середній бал за результатами сесії (дійсне число), а також про розмір стипендії (дійсне число). За умови, що у групі навчається не більше 40 студентів, описати у програмі масив для зберігання згаданої інформації, розрахувати середній бал кожного зі студентів, які склали всі іспити з оцінками, вищими за 2, і виконати нарахування стипендії усім студентам бюджетної форми навчання із середнім балом 4.0 та більше. Базова стипендія становить 600 грн. Студентам, які отримали максимальні бали з усіх іспитів, нараховується стипендія, на 25 % вища за базову стипендію. Вивести інформацію про

студентів, що мають академічні заборгованості, а також інформацію про нарахування стипендій.

## 5. КОНТРОЛЬНІ ЗАПИТАННЯ

1. Яке призначення типу даних, що мають тип «запис» ?
2. Як визначається тип даних «запис»?
3. Що таке «поле запису» і як здійснюється звертання до нього?
4. Яке призначення оператора **with**?
5. Як виконується оператор **with**?
6. Що таке «записи з варіантами» і як вони описуються?
7. У чому особливість записів з варіантами?

## СПИСОК ЛІТЕРАТУРИ

1. Безменов, М. І. Основи програмування в середовищі Delphi : навч. посіб. / М. І. Безменов. – Х. : НТУ «ХП», 2010. – 608 с.
2. Кэнтю, М. Delphi 7 : Для профессионалов / М. Кэнтю – СПб. : Питер, 2004. – 1101 с.
3. Архангельский, А. Я. Программирование в Delphi 6 / А. Я. Архангельский. – М. : БИНОМ, 2002. – 1120 с.
4. Дарахвелидзе, П. Г. Программирование в Delphi 7 / П. Г. Дарахвелидзе, Е. П. Марков. – СПб. : БХВ-Петербург, 2003. – 784 с.
5. Культин, Н. Б. Основы программирования в Delphi 7 / Н. Б. Культин. – СПб. : БХВ-Петербург, 2003. – 608 с.
6. Пестриков, В. М. Delphi на примерах / В. М. Пестриков, А. Н. Маслобоев. – СПб. : БХВ-Петербург, 2005. – 496 с.
7. Ремкеев, А. А. Курс Delphi для начинающих. Полигон нестандартных задач / А. А. Ремкеев, С. В. Федотова. – М. : СОЛОН-Пресс, 2006. – 360 с.
8. Митчелл, К. Керман. Программирование и отладка в Delphi : учебный курс / Митчелл К. Керман. – М. : Вильямс, 2004. – 720 с.
9. Парижский, С. М. Delphi : Только практика / С. М. Парижский. – К. : МК-Пресс, 2005. – 208 с.
10. Культин, Н. Б. Основы программирования в Delphi 2007 / Н. Б. Культин. – СПб. : БХВ-Петербург, 2008. – 480 с.

Навчальне видання

Методичні вказівки  
до лабораторної роботи  
«Комбінований тип даних у Delphi»  
з курсу «Програмування» для студентів напрямку 6.040302 – Інформатика  
(спеціалізація «Соціальна інформатика»)

Укладач БЕЗМЕНОВ Микола Іванович

Відповідальний за випуск О. С. Куценко  
Роботу до видання рекомендував О. В. Горелий

За авторською редакцією

План 2011 р., поз. 9 / 71-11

Підп. до друку 23.05.2011 р. Формат  $60 \times 84 \frac{1}{16}$ . Папір офісний.  
Riso-друк. Гарнітура Таймс. Ум. друк. арк. 0,6. Наклад 50 прим.  
Зам. № 165. Ціна договірна.

---

Видавничий центр НТУ «ХП».  
Свідоцтво про державну реєстрацію ДК № 3657 від 24.12.2009 р.  
61002, Харків, вул. Фрунзе, 21

---

Друкарня НТУ «ХП», 61002, Харків, вул. Фрунзе, 21